基于粒子群算法的三维 CAD 模型相似性计算 *

高雪瑶 [a]，陈育南 [a]，张春祥 [b]

(哈尔滨理工大学 a. 计算机科学与技术学院; b. 软件学院, 哈尔滨 150080)

摘　要：为了更加准确地度量两个模型之间的形状差异，提出了一种基于粒子群的模型相似性计算方法。利用面的组成边数来构造面相似度矩阵，通过粒子群算法对该矩阵进行搜索，得到了两个模型之间的最优面匹配序列。根据这个最优面匹配序列，从面相似度矩阵中提取对应的面相似性值。通过累积面之间的相似性来计算模型之间的整体相似性。以此为基础来度量模型之间的差异。实验结果表明：该方法能够准确地度量两个模型之间的相似程度。

关键词：形状差异；模型相似性；粒子群算法；匹配序列

中图分类号：TP391.7　　doi: 10.3969/j.issn.1001-3695.2017.09.0961

Similarity calculation of 3D CAD model based on particle swarm algorithm

Gao Xueyao[a], Chen Yunan[a], Zhang Chunxiang[b]

(*a. School of Computer Science & Technology, b. School of Software, Harbin University of Science & Technology, Harbin* 150080, *China*)

Abstract: In order to measure shape difference of two models accurately, this paper proposed a method for calculating model similarity based on particle swarm. It used edge number of face to construct face similarity matrix, and used particle swarm algorithm to search this matrix to get an optimal face matching sequence between two models. It extracted correspondent face similarity values according to this face matching sequence from face similarity matrix. This paper accumulated face similarities to calculate two models' similarity and measured shape difference of two models. Experimental results show that this method can evaluate similarity between two models accurately.

Key Words: shape difference; model similarity; particle swarm algorithm; matching sequence

0　引言

在三维 CAD 模型检索中，模型相似性计算起着重要的作用。它关系到模型检索的效率和可靠性。刘志通过获取三维模型较优视点集，来渲染三维模型的混合轮廓线视图。利用提取的 Gabor 边缘响应特征，去构建特征库。从特征库中，使用视觉词袋方法来检索相似的模型[1]。范菁采用多层次局部视图对树木模型进行组合描述，根据视图所对应的树木语义信息，来实现视图特征的转换，从而完成由粗略到精细的多阶段检索[2]。庄廷建立了距离—余弦二维网格，通过获取网格中的随机点数量，来构建三维模型的距离-余弦联合形状分布矩阵。利用分布矩阵之间的 L_2 距离来表示模型之间的相似度，并采用二进制粒子群优化算法来计算模型之间的相似性[3]。张开兴将两个自由曲面的相似性评价问题转换为两个曲面局部特征之间的相似性评价问题。利用自由曲面参数域内生成的曲率云图来提取类 Sift 算子，通过比较局部特征的相似程度来评价自由曲面之间的相似性[4]。白静将各类非线性特征统一表示为属性图，利用非线性凝聚层次聚类算法来实现属性图聚类。同时，引入增量模型的动态归类方法，来实现三维 CAD 模型可重用区域的有效聚类[5]。Qin 利用分层特征本体和本体映射，来生成语义描述符。以语义描述符为基础进行本体推理，使检索系统获得更好的性能[6]。Kim 给出了一种基于自由草图的建模方法。利用形状分布来比较多分辨率模型，使用简单的形状查询模型来提高检索精度[7]。Chen 给出了三个组合特征描述符和一个基于类的特征描述符，并将其应用于模型检索过程中[8]。Tao 把面邻接图分割为凸、凹和平面区域，利用区域属性编码来表示面区域，通过比较区域属性编码来度量模型之间的相似性[9]。Jin 通过推导拉伸形变能量的三维版本，结合固定边界条件，将表面参数化扩展到体积参数化，使其具有拉伸最小化的优点[10]。皇甫中民从模型的 B-Rep 表示中提取几何拓扑属性。利用属性邻接图来表示三维 CAD 模型，使用图谱来描述局部特征。同时，采用两层搜索机制来实现模型检索[11]。陈龙利用单元之间的拓扑信息，生成多个零亏格四边子域，并使用边界插值算法和内点优化算法来实现子域参数化[12]。潘万彬以区域之间的对应面和三

基金项目：国家自然科学基金资助项目（61502124）；黑龙江省自然科学基金资助项目（F201420）

作者简介：高雪瑶（1979-），女，黑龙江哈尔滨人，副教授，硕导，博士，主要研究方向为计算机图形学与 CAD（xueyao_gao@163.com）；陈育南（1993-），男，硕士研究生，主要研究方向为计算机图形学与 CAD；张春祥（1974-），男，教授，硕导，博士，主要研究方向为计算机图形学与自然语言处理.

维尺寸约束图为基础，将简单局部区域的形状和定位信息传播到可重用区域[13]。

本文利用面组成边数差异，来构建源模型与目标模型之间的面相似度矩阵。以面相似度矩阵为基础，使用粒子群算法来搜索源模型与目标模型之间的最优面匹配序列。以此为基础，计算源模型和目标模型之间的相似性。

1　源模型与目标模型之间的面相似性计算

模型是由多个面组成的。组成面的形状差异，造成了模型形状之间千差万别。在计算源模型与目标模型之间的相似性时，需要累积这两个模型面之间的相似性。如果模型之间面的组成边数差距很小，那么这两个模型的相似度就会很高；如果模型之间面的组成边数差异很大，那么这两个模型之间的相似度就很低。源模型 A 和目标模型 B 如图 1 所示。

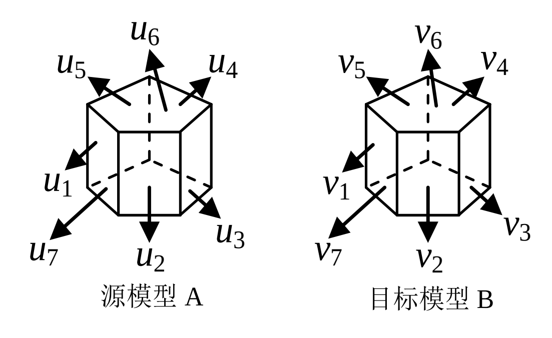


图 1　源模型 A 和目标模型 B

源模型 A 包括七个面：面 u_1、u_2、u_3、u_4、u_5、u_6 和 u_7。面 u_1 与面 u_2、u_5、u_6 和 u_7 相邻接；面 u_2 与面 u_1、u_3、u_6 和 u_7 相邻接；面 u_3 与面 u_2、u_4、u_6 和 u_7 相邻接；面 u_4 与面 u_3、u_5、u_6 和 u_7 相邻接；面 u_5 与面 u_1、u_4、u_6 和 u_7 相邻接；面 u_6 与面 u_1、u_2、u_3、u_4 和 u_5 相邻接；面 u_7 与面 u_1、u_2、u_3、u_4 和 u_5 相邻接。面 u_1 有四个相邻面，因此，面 u_1 的边数为 4。同理，可以计算出其他面的边数。面 u_2、u_3、u_4 和 u_5 分别有四条边，面 u_6 和 u_7 分别有五条边。

目标模型 B 包括七个面：面 v_1、v_2、v_3、v_4、v_5、v_6 和 v_7。其中，面 v_1、v_2、v_3、v_4 和 v_5 是四边形，分别有四条边；面 v_6 和 v_7 是五边形，分别有五条边。

利用式(1)来计算源模型面 u_i 和目标模型面 v_j 之间的相似度 $S(u_i, v_j)$。

$$S(u_i,v_j)=1-\frac{|num(u_i)-num(v_j)|}{\max(num(u_i),num(v_j))} \tag{1}$$

其中：$num(f)$表示模型面 f 所包含的边数；$max(x, y)$表示 x 与 y 之间的最大值。由公式(1)可以看出：若两个模型面所含的边数差异越小，则这两个面的形状就越相似。使用 $S(u_i, v_j)$来构造源模型 A 与目标模型 B 之间的面相似度矩阵 $\boldsymbol{S}_{AB}$，其中：m 表示源模型面数，n 表示目标模型面数。如果 m 大于 n，则将源模型与目标模型互换。即始终保持面相似度矩阵的行数要小于等于列数。

$$S_{AB}=\begin{bmatrix} & v_1 & \cdots & v_j & \cdots & v_n \\ u_1 & S(u_1,v_1) & \cdots & \cdots & \cdots & S(u_1,v_n) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ u_i & \cdots & \cdots & S(u_i,v_j) & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ u_m & S(u_m,v_1) & \cdots & \cdots & \cdots & S(u_m,v_n) \end{bmatrix}$$

其中：源模型的面序列号 u_1、u_2、…、u_m，作为面相似度矩阵的行标；目标模型的面序列号 v_1、v_2、…、v_n，作为面相似度矩阵的列标。在面相似度矩阵 $\boldsymbol{S}_{AB}$ 中，源模型面 u_i 和目标模型面 v_j 之间的相似度值为 $S(u_i, v_j)$。

源模型面与目标模型面之间的相似程度影响着两个模型之间的相似性。对于两个模型而言，如果它们的组成面越相似，那么这两个模型之间的相似度也就越高。如果它们的组成面差异越大，则这两个模型之间的相似度也就越低。

根据式(1)，构造源模型 A 与目标模型 B 之间的面相似度矩阵。在面相似度矩阵中，用源模型 A 的面序号 u_1、u_2、…、u_7 表示行号，用目标模型 B 的面序号 v_1、v_2、…、v_7 表示列号。源模型 A 和目标模型 B 的面相似度矩阵如下所示：

$$\begin{bmatrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \\ u_1 & 1 & 1 & 1 & 1 & 1 & 0.8 & 0.8 \\ u_2 & 1 & 1 & 1 & 1 & 1 & 0.8 & 0.8 \\ u_3 & 1 & 1 & 1 & 1 & 1 & 0.8 & 0.8 \\ u_4 & 1 & 1 & 1 & 1 & 1 & 0.8 & 0.8 \\ u_5 & 1 & 1 & 1 & 1 & 1 & 0.8 & 0.8 \\ u_6 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 & 1 & 1 \\ u_7 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 & 1 & 1 \end{bmatrix}$$

在源模型 A 中，面 u_1 与面 u_2、u_5、u_6 和 u_7 相邻接；在目标模型 B 中，面 v_1 与面 v_2、v_5、v_6 和 v_7 相邻接。由式(1)可知，在源模型 A 和目标模型 B 的面相似度矩阵中，第 u_1 行第 v_1 列的值为 1。在目标模型 B 中，面 v_7 与面 v_1、v_2、v_3、v_4 和 v_5 相邻接。由式(1)可知，在源模型 A 和目标模型 B 的面相似度矩阵中，第 u_1 行第 v_7 列的值为 0.8。

2　基于粒子群算法的模型面匹配

如果两个模型的组成面越相似，则这两个模型之间的相似度也就越高。对于两个模型而言，其组成面之间存在着多种对应方案。本文以面相似度矩阵为基础，使用粒子群算法来搜索源模型面与目标模型面之间的最优匹配序列。根据最优面匹配序列，来计算两个模型之间的相似性。

在使用粒子群算法搜索面相似度矩阵的过程中，设定一个 d 维的搜索空间。种群的大小为 p；x_t 表示第 t 个粒子的位置，即在面相似度矩阵中搜索到的列序号组成的向量；v_t 表示第 t 个粒子的飞行速度；在第 k 次迭代过程中，使用 $x^k_{Pbest,t}$ 表示第 t 个粒子的个体最优位置，对应的适应度函数值为 $f_{Pbest,t}$；利用 x^k_{Gbest} 表示全局最优位置，对应的适应度函数值为 f_{Gbest}。利用

$x_{Pbest,t}^{k}$ 和 x_{Gbest}^{k} 来更新 x_t^{k+1} 和速度 v_t^{k+1}，其计算过程如式(2)和(3)所示。

$$v_t^{k+1} = w * v_t^k + c_1 r_1 (x_{Pbest,t}^k - x_t^k) + c_2 r_2 (x_{Gbest}^k - x_t^k) \quad (2)$$

$$x_t^{k+1} = x_t^k + v_t^{k+1} \quad (3)$$

其中：k 表示当前的迭代次数；w 表示惯性权重；c_1、c_2 为学习因子；r_1、r_2 表示取值在[0, 1]内的随机数。

在每次迭代过程中，为了使粒子向最好的位置进行偏移，为每个粒子设定了对应的适应度值。第 t 个粒子在面相似度矩阵 $\boldsymbol{S}_{AB}$ 中搜索到的位置向量 $x_t=(j(1), j(2), ..., j(m))$。其中，$j(i)$表示与源模型 A 的第 i 个面所匹配的目标面号(i=1, 2, ..., m)。第 t 个粒子的适应度函数的计算过程如式(4)所示。

$$f(t) = \sum_{i=1}^{m} S[i, j(i)] \quad (4)$$

基于粒子群算法的面匹配过程如下所示：

a)根据式(1)构造源模型 A 和目标模型 B 之间的面相似度矩阵 $\boldsymbol{S}_{AB}$。

b)初始化种群大小 p，惯性权重 w，迭代次数 s，学习因子 c_1、c_2；初始化粒子的位置 x_t 和速度 v_t (t=1, 2, ..., n)；初始化当前迭代次数 k=0。

c)根据式(4)计算第 t 个粒子的适应度函数值 $f(t)$(t=1, 2, ..., n)。

d)判断 $k>s$，如果为真则算法结束，输出 x_{Gbest}^k；否则执行步骤 e)。

e)如果 $f(t)<f_{Pbest,t}$，则 $f_{Pbest,t}=f(t)$，$x_{Pbest,t}^k = x_t^k$。

f)如果 $f_{Pbest,t}<f_{Gbest}$，则 $f_{Gbest}=f_{Pbest,t}$，$x_{Gbest}^k = x_{Pbest,t}^k$。

g)根据式(2)(3)来更新第 t 个粒子的速度 v_t^{k+1} 和位置 x_t^{k+1}。

h)k=k+1，执行步骤 d)。

在模型面相似度矩阵 $\boldsymbol{S}_{AB}$ 中，利用粒子群算法来搜索两个模型之间的最优面匹配序列。最优位置解向量为($j(1)$, $j(2)$, ..., $j(m)$)。可以知道：源模型 A 与目标模型 B 的最优面匹配序列为(1, $j(1)$)、(2, $j(2)$)、...、(m, $j(m)$)。

根据最优面匹配序列，从源模型 A 与目标模型 B 的面相似度矩阵 $\boldsymbol{S}_{AB}$ 中提取对应的元素。通过累积源模型面和目标模型面之间的相似性，就可以得到 A、B 两个模型的整体相似性 $S_{mod(A,B)}$。模型相似性 $S_{mod(A,B)}$的计算过程如式(5)所示。

$$S_{\mathrm{mod}(A,B)} = \frac{\sum_{i=1}^{m} S[i, j(i)]}{\min(m,n)} \quad (5)$$

其中:(i, j(i))表示最优面匹配序列中的第 i 个值；$S[i,j(i)]$为面相似度矩阵 $\boldsymbol{S}_{AB}$ 的第 i 行第 j(i)列的数值；$min(m, n)$为 m, n 的最小值。

3 实验

本文在处理器为 i5 的 Windows 7 操作系统下，使用 Matlab 来编写粒子群算法，实现对面相似度矩阵的搜索。在实验中，共选取三个 CAD 模型。目标模型和源模型如图 2 所示。目标模型是四棱锥；源模型 A 是四棱锥；源模型 B 是三棱柱；源模型 C 是带有通孔的圆台。

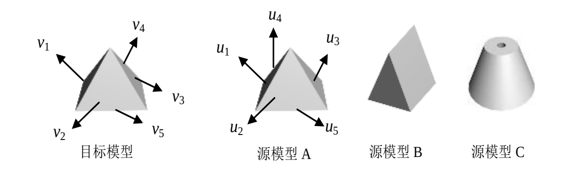


图 2 目标模型和源模型

在度量源模型与目标模型之间的相似性时，一般都采用贪心策略来寻找这两个模型之间的最优面匹配序列。为了验证本文所提出方法的有效性，共进行了两组对比实验。

在实验中，利用式(1)来构造源模型与目标模型的面相似度矩阵 $\boldsymbol{S}_{AB}$。分别使用贪心算法和粒子群算法来搜索这两个模型之间的最优面匹配序列。根据最优面匹配序列，从矩阵 $\boldsymbol{S}_{AB}$ 中提取源模型面与目标模型面之间的相似性，并使用式(5)来计算模型之间的相似性。

目标模型包括五个面：面 v_1、v_2、v_3、v_4 和 v_5。面 v_1 与面 v_2、v_4 和 v_5 相邻接；面 v_2 与面 v_1、v_3 和 v_5 相邻接；面 v_3 与面 v_2、v_4 和 v_5 相邻接；面 v_4 与面 v_1、v_3 和 v_5 相邻接；面 v_5 与面 v_1、v_2、v_3 和 v_4 相邻接。面 v_1 有三个相邻面，因此，面 v_1 的边数为 3。同理，可以计算出其他面的边数。面 v_2、v_3 和 v_4 分别有三条边，面 v_5 有四条边。

以图 2 中的源模型 A 为例，使用粒子群算法来搜寻源模型 A 和目标模型之间的最优面匹配序列。源模型 A 包括五个面：面 u_1、u_2、u_3、u_4 和 u_5。其中，面 u_1、u_2、u_3 和 u_4 为三角形，分别有 3 条边；面 u_5 是四边形，含有四条边。

使用式(1)计算源模型面与目标模型面之间的相似性，并构造源模型 A 与目标模型之间的面相似度矩阵，如下所示：

$$\begin{bmatrix} & v_1 & v_2 & v_3 & v_4 & v_5 \\ u_1 & 1 & 1 & 1 & 1 & 0.75 \\ u_2 & 1 & 1 & 1 & 1 & 0.75 \\ u_3 & 1 & 1 & 1 & 1 & 0.75 \\ u_4 & 1 & 1 & 1 & 1 & 0.75 \\ u_5 & 0.75 & 0.75 & 0.75 & 0.75 & 1 \end{bmatrix}$$

在源模型 A 中，面 u_1 与面 u_2、u_4 和 u_5 相邻接；在目标模型中，面 v_1 与面 v_2、v_4 和 v_5 相邻接。由式(1)可知,在源模型 A 和目标模型的面相似度矩阵中，第 u_1 行第 v_1 列的值为 1。在目标模型中，面 v_5 与面 v_1、v_2、v_3 和 v_4 相邻接。由式(1)可知,在源模型 A 和目标模型的面相似度矩阵中，第 u_1 行第 v_5 列的值为

0.75。

以源模型 A 和目标模型的面相似度矩阵为基础，使用粒子群算法来搜索源模型 A 与目标模型之间的最优面匹配序列。得到最优面匹配序列为：(1, 3)、(2, 2)、(3, 1)、(4, 4)、(5, 5)。同时，使用式(5)来计算源模型 A 与目标模型之间的整体相似性，其数值为 1。

源模型 A、源模型 B 和源模型 C 与目标模型之间的相似性计算结果如表 1 所示。

表 1　源模型与目标模型之间的相似性

源模型	面数	边数	顶点数	贪心算法	粒子群算法
A	5	8	5	1	1
B	5	9	6	0.9	0.9
C	4	8	0	0.75	0.8125

从形状上看，源模型 A 与目标模型完全一致，其相似性数值为 1；源模型 B 与目标模型的相似度较大，其相似性数值为 0.9；源模型 C 与目标模型有一定的相似性，其相似性数值为 0.812 5。

从表 1 可知,源模型 A 和目标模型是同一个模型，贪心算法和粒子群算法获得的数据是一致的。对于源模型 B，贪心算法和粒子群算法获得的数据也是一致的。对于源模型 C，它是一个关键性模型。使用粒子群算法所获得的相似性数值比贪心算法要高，其结果提高了 8.33%。从形状上看，源模型 C 与目标模型有一定的相似性。相对于贪心算法而言，本文所提出的方法更能准确地度量两个模型之间的相似性。

在实验中，使用贪心算法搜索面相似度矩阵，其时间复杂度为 $O(mn)$。使用粒子群算法搜索面相似度矩阵，其时间复杂度为 $O(psmn)$。尽管粒子群的时间复杂度要高于贪心算法，但是粒子群算法在度量模型之间的相似性时，效果要好于贪心算法。

4　结束语

本文利用面的边数差异来构建面相似度矩阵。通过粒子群算法来搜索源模型与目标模型之间的最优面匹配序列。以最优面匹配序列为基础，通过累积源模型面与目标模型面之间的相似性来度量两个模型的整体相似度。实验结果表明：粒子群算法可以有效地实现源模型面与目标模型面的匹配，进而准确地度量模型之间的形状差异。

参考文献：

[1] 刘志，尹世超，潘翔，等. 基于特征线条的三维模型检索方法 [J]. 计算机辅助设计与图形学学报, 2016, 28 (9): 1512-1520.

[2] 范菁，李然，董天阳. 基于局部视图的三维树木模型递进检索方法 [J]. 计算机辅助设计与图形学学报, 2016, 28 (1): 162-171.

[3] 庄廷，张旭堂，侯珍秀. 多特征结合相似度优化的三维工程模型检索算法 [J]. 哈尔滨工程大学学报, 2015, 36 (5): 720-724.

[4] 张开兴，白晓亮，张树生. 基于局部形状特征的自由曲面相似性评价方法 [J]. 计算机集成制造系统, 2014, 20 (3): 530-536.

[5] 白静，罗皓楠，秦飞巍. 面向非线性特征的三维 CAD 模型聚类 [J]. 计算机辅助设计与图形学学报, 2015, 27 (8): 1578-1586.

[6] Qin Feiwei, Gao Shuming, Yang Xiaoling, et al. An ontology-based semantic retrieval approach for heterogeneous 3D CAD models [J]. Advanced Engineering Informatics, 2016, 30 (4): 751-768.

[7] Kim Hyungki, Cha Moohyun, Mun Duhwan. Shape distribution based retrieval of 3D CAD models at different levels of detail [J]. Multimedia Tools and Applications, 2017, 76 (14): 15867-15884.

[8] Chen Qiang, Fang Bin, Yu Yongmei, et al. 3D CAD model retrieval based on the combination of features [J]. Multimedia Tools and Applications, 2015, 74 (13): 4907-4925.

[9] Tao Songqiao, Wang Shuting, Chen Anhui. 3D CAD solid model retrieval based on region segmentation [J]. Multimedia Tools and Applications, 2017, 76 (1): 103-121.

[10] Jin Yao, Qian Gui Ping, Zhao Jie Yi, et al. Stretch-minimizing volumetric parameterization [J]. Journal of Computer Science and Technology, 2015, 30 (3): 553-564.

[11] 皇甫中民，张树生，闫雒恒. 基于层次特征描述子的三维 CAD 模型检索 [J]. 计算机集成制造系统, 2015, 21 (12): 3095-3106.

[12] 陈龙，樊兴旺，王猛，等. 有限元四边单元网格模型的参数化重建 [J]. 计算机辅助设计与图形学学报, 2017, 29 (4): 680-688.

[13] 潘万彬，高曙明，陈翔. 参数化模型的局部自动适应重用方法 [J]. 计算机辅助设计与图形学学报, 2016, 28 (2): 314-327.